

3.4.4 情報システム連携の枠組み構築

(1) 業務の内容

(a) 業務の目的

広域連携における情報共有を実現するには、既存システムや新規システムなど多種多様な情報システムを相互に連携させていかなければならない。このためには、情報共有システム連携の枠組みの構築が不可欠となる。

これまで開発してきた減災情報共有プロトコル(MISP)および減災情報共有データベース(DaRuMa)を基盤として、広域連携に必要な機能を洗い出し、プロトコルの再設計およびそれに基づいたデータベースプロトタイプシステムの拡張を行う。また、「広域連携のための情報コンテンツの構築」で整理された情報コンテンツを分析し、多数のコンテンツの情報表現を記述するスキーマの作成・管理を行うためのシステムを開発すると同時に、基本的なデータ構造についての設計・標準化を行い、プロトコル等の情報共有データベースの枠組みを完成させる。

さらに、実証実験のために必要な接続ツール群を拡充し、情報共有データベースプロトタイプシステムの構築を行って、実証実験に参画し、これらのシステムの評価を行う。

(b) 平成 21 年度業務目的

平成 20 年度までに拡張したデータベースプロトタイプシステム(DaRuMa)に、広域連携の評価実験（システム連携のデモ）のためのデータを登録し、実験に参画する。そのために、個別研究テーマ（2）の他グループで開発するシステムとの連携ツールの開発および改良を進める。また、アメリカで開催される OMG Technical Meeting に出席し、地理データ標準および DaRuMa の国際標準化に向けた調査を行う。

(c) 担当者

所属機関	役職	氏名	メールアドレス
独立行政法人産業技術研究所 情報技術研究部門	主任研究員	野田五十樹	
同	テクニカルスタッフ	下羅弘樹	

(2) 平成 21 年度の成果

(a) 業務の要約

本年度については、前年度までに拡張した DaRuMa による広域連携の評価実証のため、個別研究テーマの他グループで開発するシステムとの連携ツールの開発および改良を進め、12 月および 3 月の実証デモに参画した。

また、国際標準化団体の 1 つである OMG Technical Meeting に出席し、地理データ標準および DaRuMa の国際標準化に向けた調査を行なった。

(b) 業務の成果

1) 実証デモシステムの構築

本サブテーマでは、情報システム連携の枠組みの有効性を示すため、各種情報システムを DaRuMa と MISIP を介して接続して円滑な情報共有を実現する実証デモを、12月と3月に行なった。このデモにおいて接続を行なったシステムは以下の通りである。

- 東京大学生産研において設計・開発が進められている医療機関と救急搬送の管理システム
- 山梨大学において開発されている自治体向け庁内災害対策管理システム
- 消防研で開発されている延焼シミュレーションシステム
- 研究協力機関である JAXA で進められている救急ヘリのための管制システム(D-Net)
- 東京大学情報理工学系研究科で進められている汎用 Viewer

また、これら以外に、将来的に実現されると想定している通報システムからの情報付与、災害状況から危険道路を割り出す判定、既存の基盤地図情報の変換・登録などの機能を持ったツール群を作成し、ストーリーのあるデモの実現を行なった。

実証デモの接続の全体像を、図1に示す。

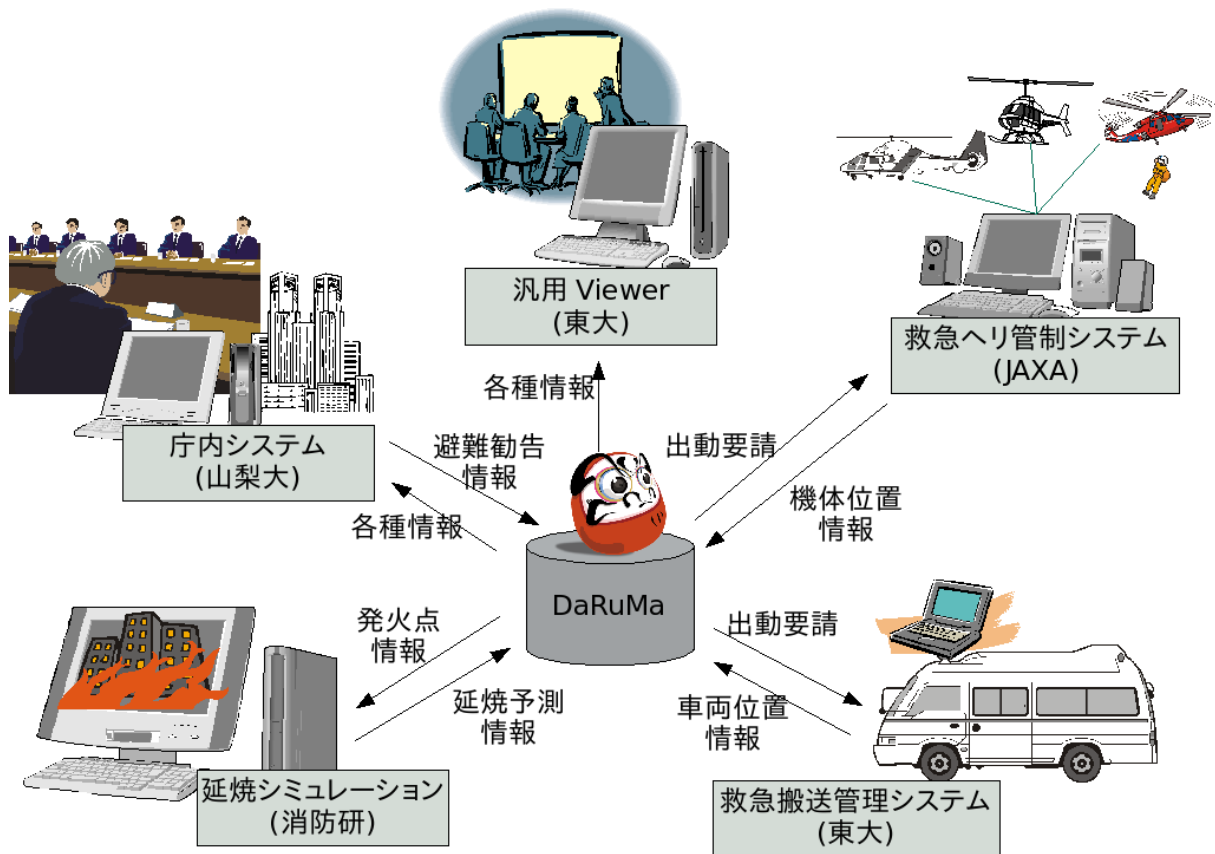


図1 実証デモにおけるシステム連携

この図に示されているように、各システムは以下のような入出力を行ない、情報共有を実現している。

- 救急搬送管理システム: 自治体からの出動要請や各種道路情報を受け、患者の配送先や経路を決定し、搬送中はその車両位置情報を DaRuMa に登録することで、他機関との連携を

円滑にする。

- 庁内システム: 自治体で収集される災害情報や DaRuMa に集約される各種情報を受け、避難勧告などを決定、それらの広報情報を DaRuMa に登録し、関係機関との連携をはかる。
- 延焼シミュレーションシステム: 通報やヘリからの観察による火災発生情報から火事の延焼範囲を数十時間にわたり推定し、延焼危険領域を割り出し、DaRuMa に登録する。
- D-Net: 出動要請などに応じて、各機体の位置情報やミッション遂行状況を DaRuMa に登録し、搬送先や自治体などとの円滑な連携を実現する。
- 汎用 Viewer: 自治体や医療期間・救急車で稼働することを前提として、DaRuMa に登録されているほぼ全ての情報を取得し、地図上に表示し、一覧性のある形で災害状況を提示し、人の判断を促す。

以下では、デモで用いたシナリオに基づき、実証デモで用いた各種情報表現およびシステム間での情報のやり取りを説明する。

2) 火災情報通報と延焼シミュレーション

延焼シミュレーションシステムを各種情報システムと連携させることで、危険予知などを円滑に行なえることを示すために、通報などの火災発生情報から数時間後までの各時刻における延焼を予測し、危険領域として情報提供するデモシステムを構築した。

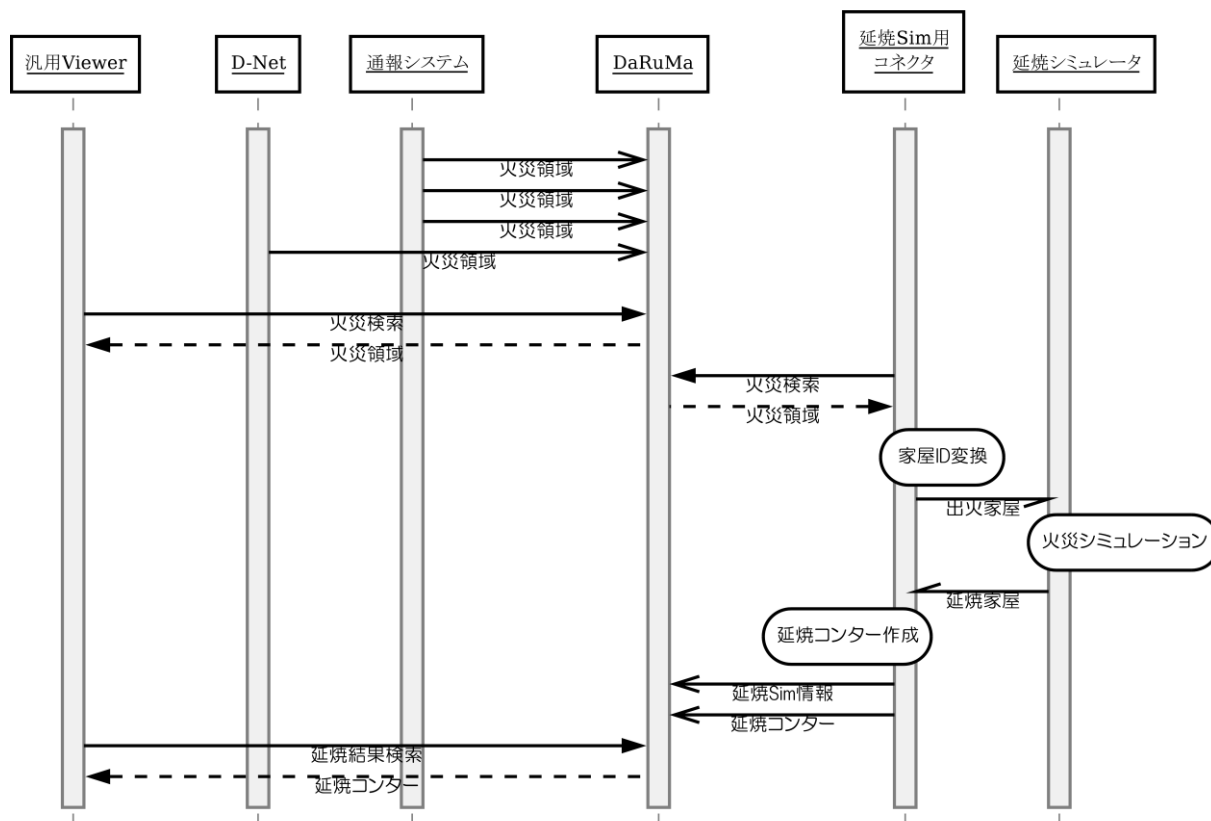


図2 火災情報通報と延焼シミュレーションのシステム連携手順

各システム間での情報の受け渡しは図2に示すような形になっている。具体的には以下のようなステップでのデモを行なった。

- まず、住民からの通報などによる火災発生情報(火災領域、 DamagedAreaReport)が DaRuMa に時々刻々と登録される。
- さらに、ヘリによる上空からの観測データにより、広域にわたる火災情報 (DamagedAreaReport)も DaRuMa に登録される。(この部分は、D-Net には具体的にそれを扱う機能はまだないため、別プログラムとして実現)
- これらの情報については、常時汎用 Viewer に送られ、現在の状況が確認できるようになっている。
- 延焼シミュレーションを行なうために、DaRuMa に登録された火災発生情報 (DamagedAreaReport)を集約する。この情報には火災領域のみが記述されているため、それを、延焼シミュレーションであつかう各家屋 ID と付き合わせ、シミュレーションの初期値となる出火家屋情報に変換する。
- 火災シミュレーションを行ない、一定間隔毎の延焼状況を計算し、延焼家屋(の ID 列)として出力する。この出力結果をもとに、延焼の領域をコンターの形で求め、延焼コンター (EnshoContour)として、延焼シミュレーション情報(FireSimulationInfo)とともに DaRuMa に登録する。
- これらの計算結果は汎用 Viewer に自動的に反映され、将来に渡る危険領域の変化として確認することができる。

この流れのプロセスで用いられる情報は以下のようにになっている。

- **DamagedAreaReport**:住民からの通報およびヘリからの観測により得られた災害被害を DaRuMa に登録する際に用いる。報告には ID(報告 ID)が一意に振られ、報告者を表す報告者 ID と報告時刻とともに、被害のタイプ、被害地域などの情報が付与される (図 3)。



図 3 災害被害通報(DamagedAreaReport)の構造

```

<DamagedAreaReport
xmlns="http://admire.or.jp/Experience/ShutoChokka/2009.09/FireSimulation">
  <reporterId>0</reporterId>
  <reportId>1</reportId>
  <reportTime>2009-12-18T10:00:00+09:00</reportTime>
  <damageType>fire</damageType>
  <location>
    <Polygon srsName="EPSG:4326" xmlns="http://www.opengis.net/gml">
      <outerBoundaryIs>
        <LinearRing>
          <coordinates>
            139.720104,35.518052 139.720104,35.51807
            139.720122,35.51807 139.720122,35.518052
            139.720104,35.518052 139.720104,35.518052
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
    </Polygon>
  </location>
</DamagedAreaReport>

```

```

    </LinearRing>
  </outerBoundaryIs>
</Polygon>
</location>
<note/>
</DamagedAreaReport>

```

- **FireSimulationInfo**: 火災延焼シミュレーションの結果のメタデータを格納する。シミュレーション毎に一意に振られたシミュレーション ID と、シミュレーションの実行時刻（開始および終了）、全延焼の boundarybox を示す area 属性と、シミュレーションの初期値として与えた発火点の数、シミュレーションの結果として得られた総延焼棟数および延焼コンターの数を情報として含んでる（図4）。

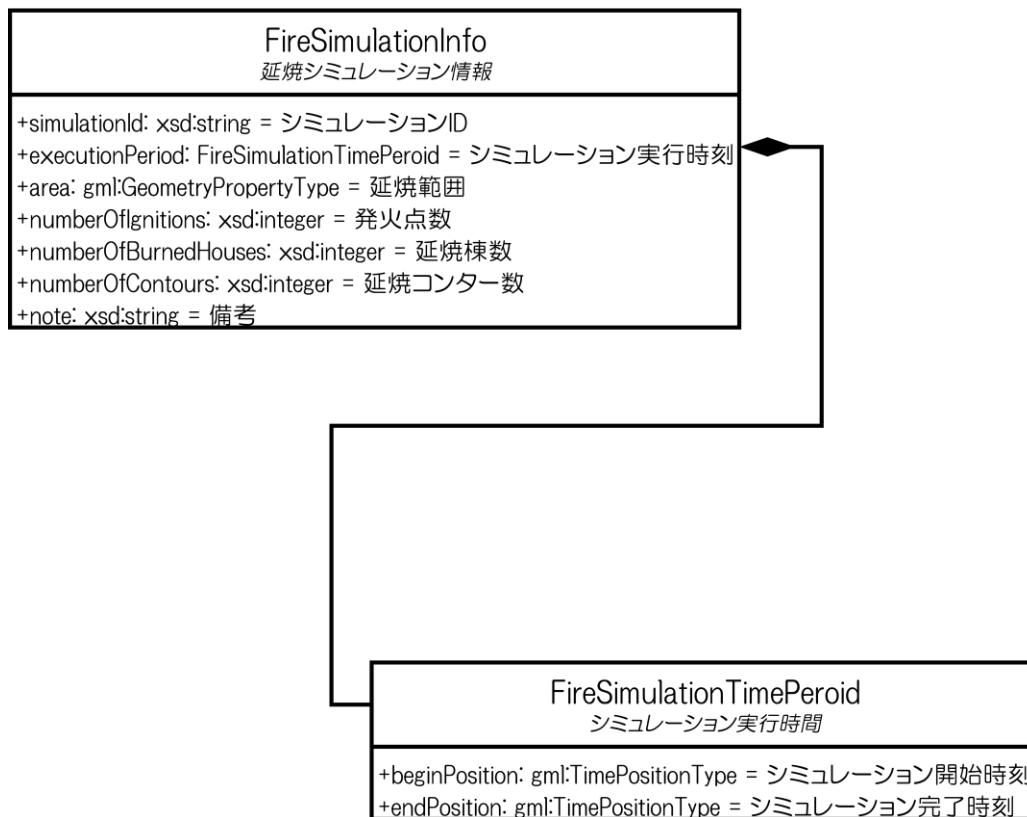


図4 延焼シミュレーションのメタ情報(FireSimulationInfo)の構造

```

<FireSimulationInfo
  xmlns="http://admire.or.jp/Experience/ShutoChokka/2009.09/FireSimulation">
  <simulationId>2009-12-18T11:24:34+9:00</simulationId>
  <executionPeriod>
    <beginPosition>2009-12-18T14:14:22+09:00</beginPosition>
    <endPosition>2009-12-18T14:14:22+09:00</endPosition>
  </executionPeriod>
  <area>
    <Box xmlns="http://www.opengis.net/gml">
      <coordinates>139.494452,35.36871 139.744037,35.621231</coordinates>
    </Box>
  </area>
  <numberOfIgnitions>1181</numberOfIgnitions>
  <numberOfBurnedHouses>43640</numberOfBurnedHouses>
  <numberOfContours>1181</numberOfContours>
  <note>File: ./Samba/ignitionData.2009-1218-141251.en sho.dat</note>

```

</FireSimulationInfo>

- EnshoContour: 火災延焼シミュレーションの結果を領域の形で格納する。対応する FireSimulationInfo のシミュレーション ID と、初期値で与えられた発火元の ID、および発火・鎮火時刻と地震発生後(火災発生後ではない)からの経過時間、延焼の全体のコンターと関係している建物の総数およびその ID リストを情報として含んでる (図5)。

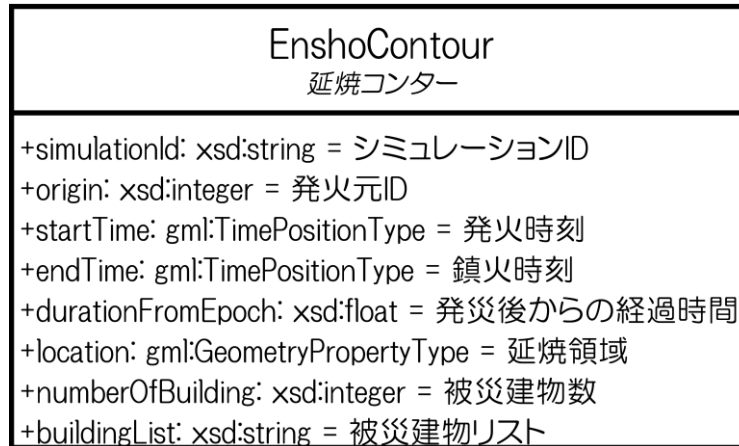


図5 延焼コンター(EnshoContour)の構造

```
<EnshoContour
xmlns="http://admire.or.jp/Experience/ShutoChokka/2009.09/FireSimulation">
  <simulationId>2009-12-18T11:24:34+9:00</simulationId>
  <origin>10003</origin>
  <startTime>2009-12-18T11:00:00+09:00</startTime>
  <endTime>2009-12-18T12:00:00+09:00</endTime>
  <durationFromEpoch>7200</durationFromEpoch>
  <location>
    <Polygon xmlns="http://www.opengis.net/gml">
      <outerBoundaryIs>
        <LinearRing>
          <coordinates>
            139.522471,35.617951 139.522701,35.618162
            139.52275,35.618268 139.522839,35.618538
            139.522774,35.618722 139.52255,35.618997
            139.522197,35.619119 139.52157,35.619098
            139.521447,35.618728 139.521478,35.61843
            139.521636,35.618125 139.522013,35.618042
            139.522318,35.617977 139.522471,35.617951
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
    </Polygon>
  </location>
  <numberOfBuilding>80</numberOfBuilding>
  <buildingList/>
</EnshoContour>
```

3) ヘリによる救急搬送

大規模災害では広域連携が重要であり、中でも広域における医療連携は人的被害の拡大を防ぐ重要な要素である。この広域医療搬送では、ヘリによる救急搬送が重要な役割を果たす。

本年度の実証デモでは、このヘリによる搬送の管理と医療機関などでの情報共有の有効性を示すことを試みた。デモにおける各システム間の情報の流れは図6に示す通りである。

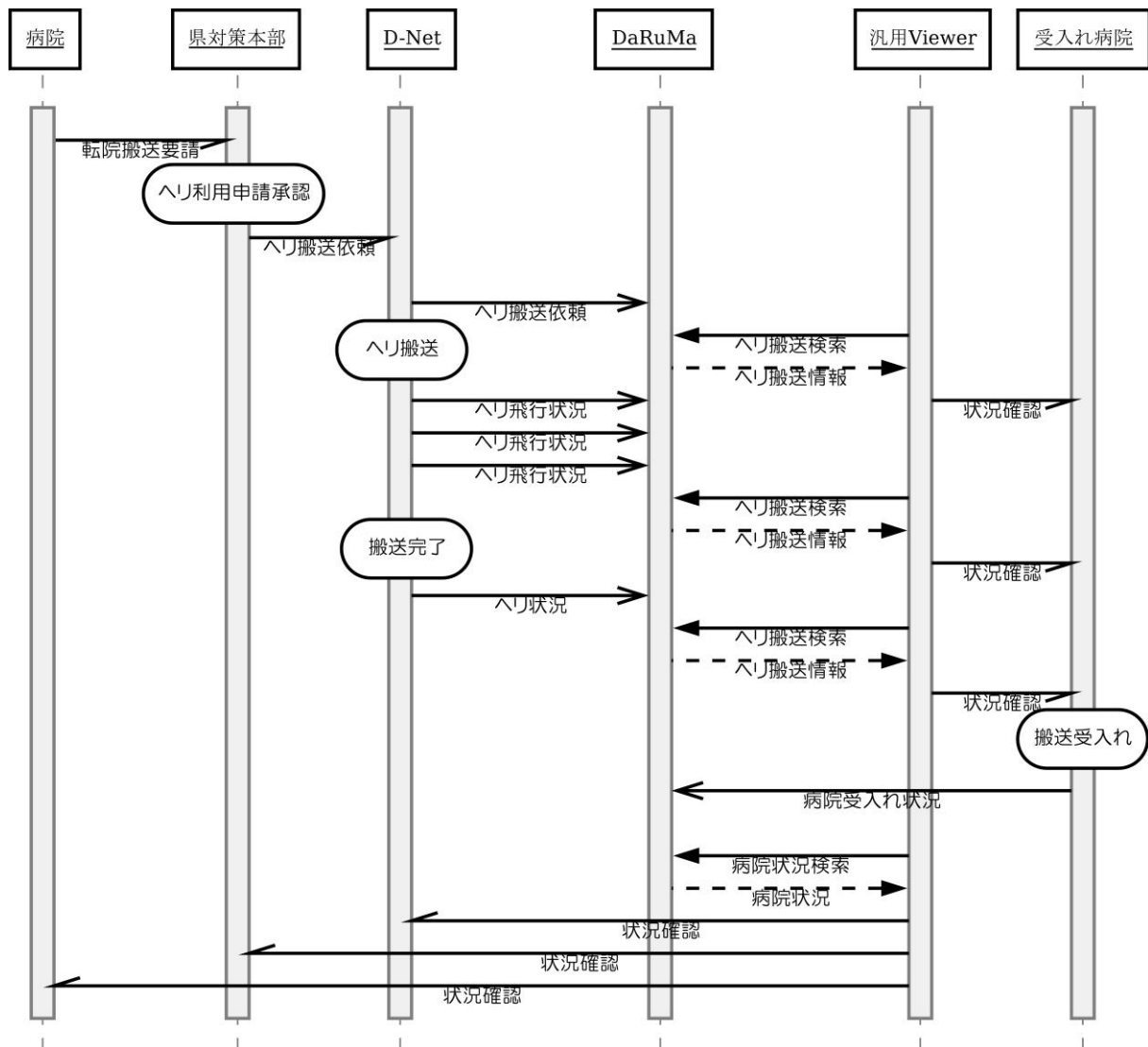


図6 ヘリによる救急搬送のシステム間の通信

具体的には以下のようなシステム連携が行なわれている。

- まず、病院から自治体の対策本部に転院搬送要請があり、ヘリによる搬送に決定された時点で、D-Net にヘリ搬送依頼が送られる。D-Net では機体の選定が行なわれ、搬送を開始する。
- 搬送開始と同時に、DaRuMa にはヘリによる搬送ミッションの情報が、ヘリの情報(AcInfo)とともに逐次書き込まれる。
- DaRuMa に書き込まれたヘリの情報はリアルタイムで汎用 Viewer に反映され、病院などで状況を確認できる。
- 無事ヘリが到着し、受け入れ病院にて搬送完了が確認されると、病院側の病床情報(HospitalInfo)が更新され、DaRuMa に登録される。この新しい病院情報は、他の機関でもDaRuMa を介して共有される。

このプロセスにおいて用いられる情報は以下のようにになっている。

- **AcInfo:** 救助ヘリの状況を表す。ヘリ毎に割り当てられた機番とともに、現在時刻とそのときの位置(緯度経度)と方位・高度および対地速度が記録される。同時に、なにかの任務を行っている場合、機体や任務の状況、および任務完了予定時刻などが記録される(図7)。

AcInfo 救助ヘリ情報
+acCode: xsd:string = 機番
+updateDate: gml:TimePositionType = 日付
+acPosition: gml:GeometryPropertyType = 緯度 経度
+direction: xsd:float = 方位
+gpsAlt: xsd:integer = 高度
+acSpeed: xsd:float = 対地速度
+acStatus: xsd:string = 体ステータス
+acmissionStatus: xsd:string = 機体任務ステータス
+acTimeOfArrival: gml:TimePositionType = 到着時刻
+missionId: xsd:integer = 任務ID
+missionType: xsd:string = 任務種別

図7 ヘリ情報(AcInfo) の構造

```
<AcInfo xmlns="http://www.jaxa.jp/dnet">
  <acCode>JA119B</acCode>
  <updateDate>2009-12-18T14:28:58.000+09:00</updateDate>
  <acPosition>
    <Point srsName="EPSG:4326" xmlns="http://www.opengis.net/gml">
      <coordinates>35.74734,139.7797</coordinates>
    </Point>
  </acPosition>
  <direction>84.87</direction>
  <gpsAlt>0</gpsAlt>
  <acSpeed>0</acSpeed>
  <acStatus>活動中</acStatus>
  <acmissionStatus>開始</acmissionStatus>
  <missionId>1701</missionId>
  <missionType>救助活動</missionType>
</AcInfo>
```

- **HospitalInfo:** 病院の病床数や受け入れ状況を表す。病院に振られた ID とともに、状況が報告された時刻と病院名、所在地の緯度経度、病院の種別が記載される。同時に、病床数や空きベッド数、重症患者数など、災害時の患者の受け入れ可能性を示す情報が含まれている(図8)。

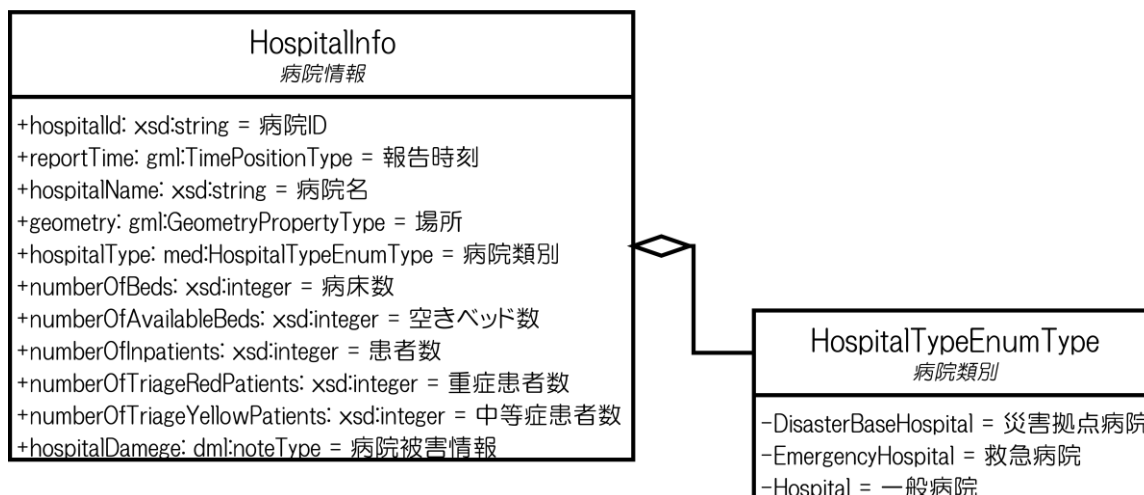


図8 病院情報(HospitalInfo)の構造

```

<HospitalInfo xmlns="http://sequoia.a01.aist.go.jp/schema/medical">
  <hospitalId>658</hospitalId>
  <reportTime>2009-12-18T14:10:00+09:00</reportTime>
  <hospitalName>川崎市立川崎病院</hospitalName>
  <geometry>
    <Point srsName="EPSG:4326" xmlns="http://www.opengis.net/gml">
      <coordinates>139.706248,35.526535</coordinates>
    </Point>
  </geometry>
  <hospitalType>DisasterBaseHospital</hospitalType>
  <numberOfBeds>733</numberOfBeds>
  <numberOfAvailableBeds>5</numberOfAvailableBeds>
  <numberOfInpatients>720</numberOfInpatients>
  <numberOfTriageRedPatients>130</numberOfTriageRedPatients>
  <numberOfTriageYellowPatients>250</numberOfTriageYellowPatients>
  <hospitalDamage>
    <text xmlns="http://www.infosharp.org/schemas/dml">大きな損傷無し</text>
  </hospitalDamage>
</HospitalInfo>
  
```

4) 避難勧告

自治体において避難勧告を決定した場合、それを効果的に広報する必要がある。本デモでは情報共有によりその広報を支援できることを示すため、自治体の庁内で意思決定などを行なうのに使われる庁内システムから、DaRuMa を介して汎用 Viewer などに勧告情報を配信することを行なった。図9はその具体的なプロセスを示している。

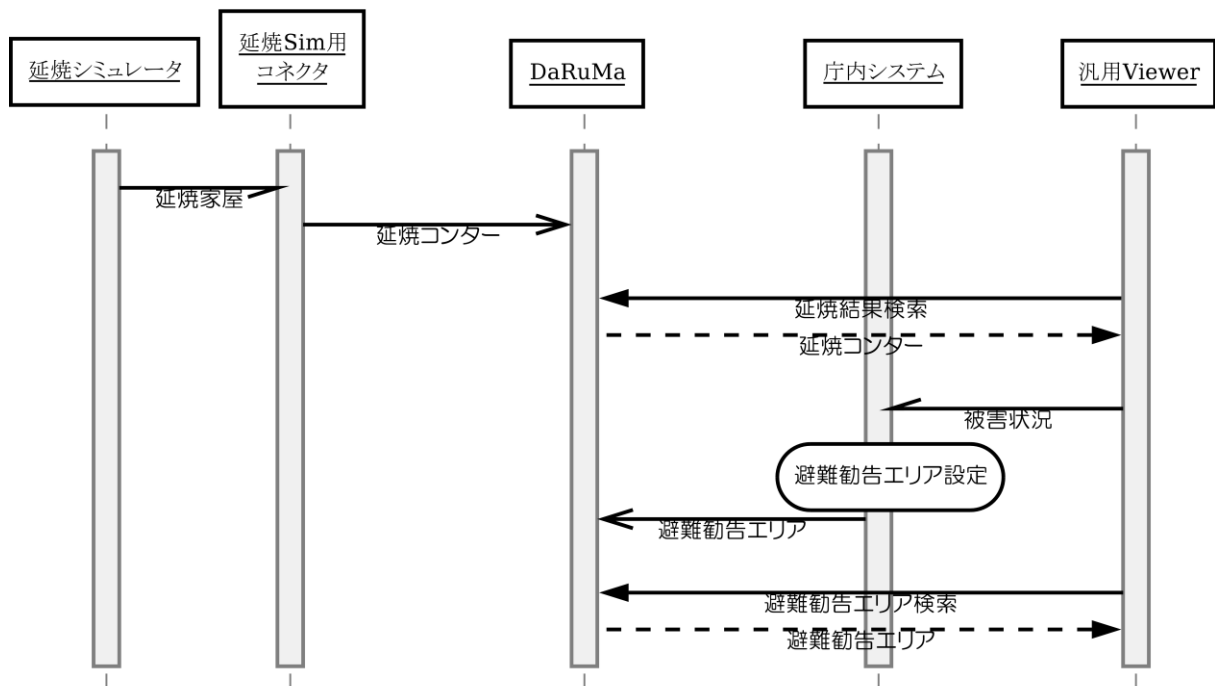


図9 避難勧告の広報

- まず各種通報やシミュレーションによる分析結果など DaRuMa に集約された情報などを参照しながら避難勧告が決定され、庁内システムで入力されたあと、勧告エリアとともに DaRuMa に登録。(なお、庁内システム側では完全に対応できていないため、支援ツールにて DaRuMa に入力)
- 登録された勧告は汎用 Viewer で確認できる。勧告エリアは領域情報として登録されているので、DaRuMa の地理検索機能により、関係する地域での勧告のみを検索・表示することもできる。

ここで用いられている避難勧告は、以下のような構造を持っている。

- **HinanKankokuHatureiInfo**: 勧告の発令内容(message)が、発令者や重要度、種別などとともに記録される (図10)。

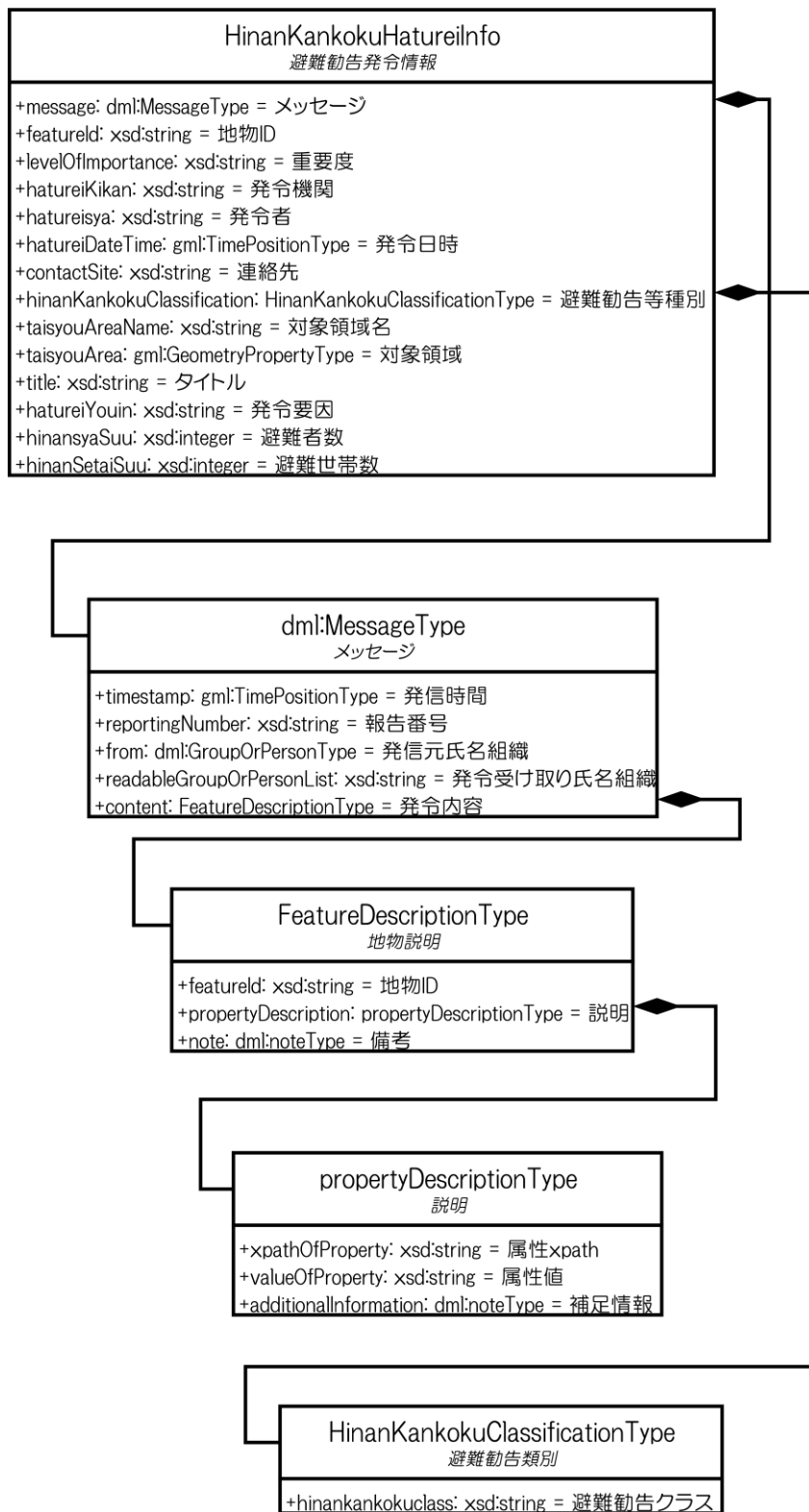


図 1 0 避難勧告情報の構造

```

<HinanKankokuHatureiInfo
  xmlns="http://www.infosharp.org/schemas/dml">
  <message>
    <timestamp>2009-12-18T11:35:00+09:00</timestamp>
    <reportingNumber>2</reportingNumber>
    <from/>
    <readableGroupOrPersonList>#all</readableGroupOrPersonList>
  
```

```

    <content>
      <featureId>HinanKankokuHatureiInfo-2</featureId>
    </content>
  </message>
  <featureId>HinanKankoku-2</featureId>
  <levelOfImportance>緊急</levelOfImportance>
  <hatureiKikan>横浜市</hatureiKikan>
  <hatureisya>鶴見区区災对本部長</hatureisya>
  <hatureiDateTime>2009-12-18T11:35:00+09:00</hatureiDateTime>
  <contactSite>横浜市</contactSite>
  <hinanKankokuClassification>
    <hinankankokuclass>避難勧告</hinankankokuclass>
  </hinanKankokuClassification>
  <taisyouAreaName>横浜市鶴見区全域</taisyouAreaName>
  <taisyouArea>
    <Polygon srsName="EPSG:4326" xmlns="http://www.opengis.net/gml">
      <outerBoundaryIs>
        <LinearRing>
          <coordinates>
            139.6386683017,35.516314524
            139.6385232392,35.5161204899
            ...
            ...
            139.6387496696,35.5164047947
            139.6386683017,35.516314524
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
    </Polygon>
  </taisyouArea>
  <title>横浜市鶴見区全域に避難勧告</title>
  <hatureiYouin>火災</hatureiYouin>
  <hinansyaSuu>400000</hinansyaSuu>
  <hinanSetaiSuu>1300000</hinanSetaiSuu>
</HinanKankokuHatureiInfo>

```

5) 救急車による搬送

ヘリによる救急搬送と同様に、救急車による搬送でも情報共有を行なうことが可能である。現在、救急車には情報端末を配備することが検討されているが、この端末を DaRuMa と連携させることで、様々な情報共有が可能となる。

デモにおいては、図 1 1 に示すような手順による情報共有を行なった。

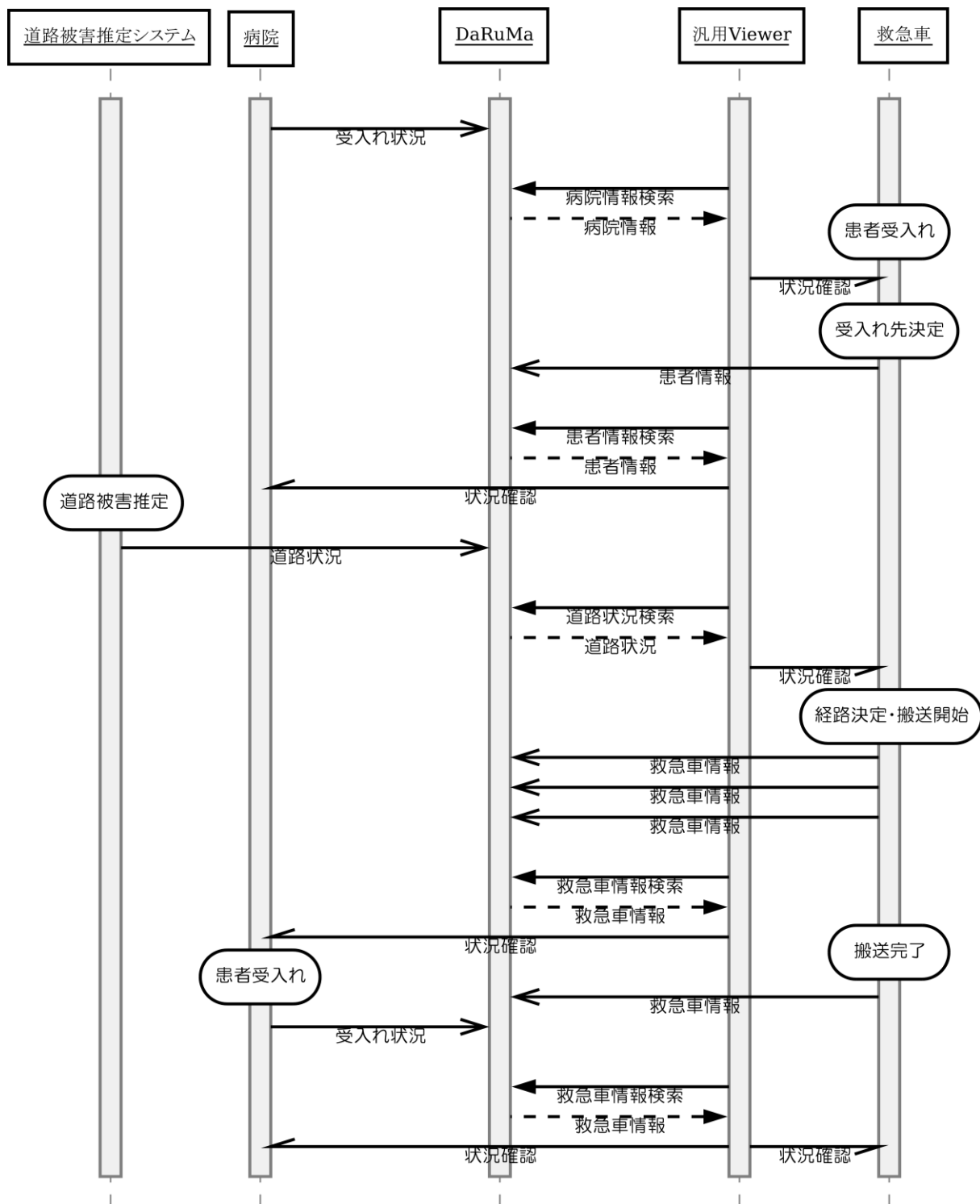


図 1 1 救急車による搬送

- ヘリと同様に搬送要請により救急車に患者が乗せられ、受け入れ病院を選定する。この際、各病院の受け入れ状況などを汎用 Viewer で確認することができる。
- また、道路被害などを各種システムから集積し、道路状況に応じて搬送経路を決定することができる。
- 搬送が開始されると、その位置情報が時々刻々 DaRuMa に登録され、汎用 Viewer などにより各機関で状況を確認できる。

- 患者を受け入れた病院では、それに応じて空き病床数などの情報を DaRuMa に反映すれば、それを関係機関で確認でき、その後の救助活動を円滑化する情報を共有できる。

このプロセスでは救急車の情報である AmbulanceInfo がやり取りされる。

- AmbulanceInfo:AcInfo と同じく、救急車の ID や時刻と現在値、現在の任務の情報などが記述される。また、患者の状況を示す写真データを格納する部分ももうけている (図 1 2)。

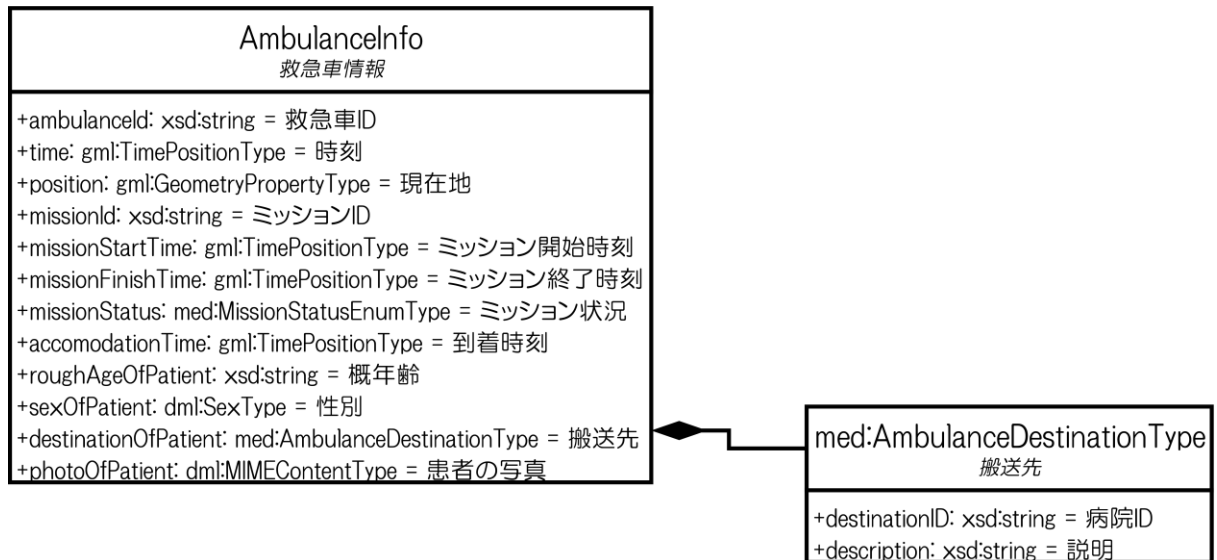


図 1 2 救急車情報(AmbulanceInfo) の構造

```

<AmbulanceInfo>
  <ambulanceId>ambulance-1</ambulanceId>
  <time>2009-12-18T13:30:00+09:00</time>
  <position>
    <gml:Point srsName="EPSG:4326">
      <gml:coordinates>139.678350,35.537800</gml:coordinates>
    </gml:Point>
  </position>
  <missionId>mission-4213</missionId>
  <missionStartTime>2009-11-20T14:53:25+09:00</missionStartTime>
  <missionFinishTime>2009-11-20T14:53:25+09:00</missionFinishTime>
  <missionStatus>InOperation</missionStatus>
  <accomodationTime>2009-11-20T14:53:25+09:00</accomodationTime>
  <roughAgeOfPatient>50~60代</roughAgeOfPatient>
  <sexOfPatient>male</sexOfPatient>
  <destinationOfPatient>
    <destinationID>hospital-658</destinationID>
    <description>川崎市立川崎病院救急科</description>
  </destinationOfPatient>
</AmbulanceInfo>
  
```

6) 道路被害推定

火災状況やその延焼推定エリアから、危険性の高い道路を割り出すことが可能である。デモにおいても、延焼シミュレーションの結果から、火災に隣接して走行車両が危険にさらされる道路を割り出し、情報を提供する実験を行なった。

推定のための具体的な手順は図 1 3 および下記の通りである。

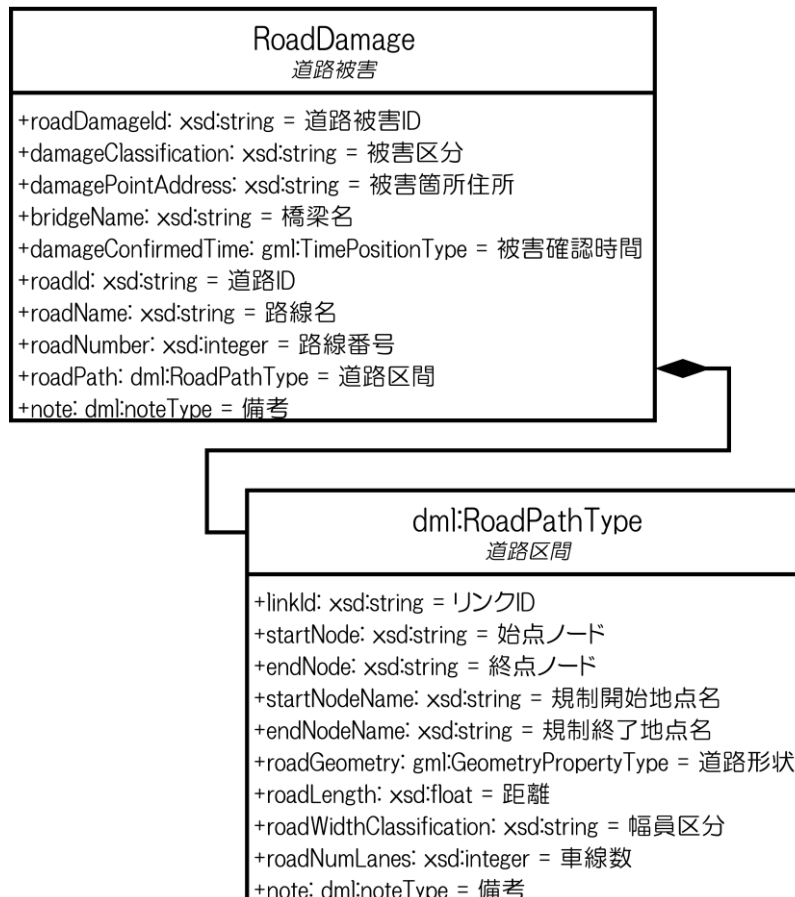


図 1 3 道路被害推定の手順

- 延焼シミュレーションの結果得られた延焼コンターを検索し、延焼エリアを求める。
- 延焼エリアごとに、それとオーバーラップするような道路(Road)を DaRuMa の地理検索機能を使って行ない、危険道路を割り出す。
- 危険と割り出された道路については、道路被害推定情報(CalculatedRoadDamage)として DaRuMa に登録する。

このプロセスで用いられているデータは以下の通りである。

- **CalculatedRoadDamage:**災害による被害をうけた、あるいは受ける可能性のある道路の情報を示す。被害について一意に振られた ID とともに、被害状況や被害を受ける道路区分(roadPath)などの情報を記述する (図 1 4)。

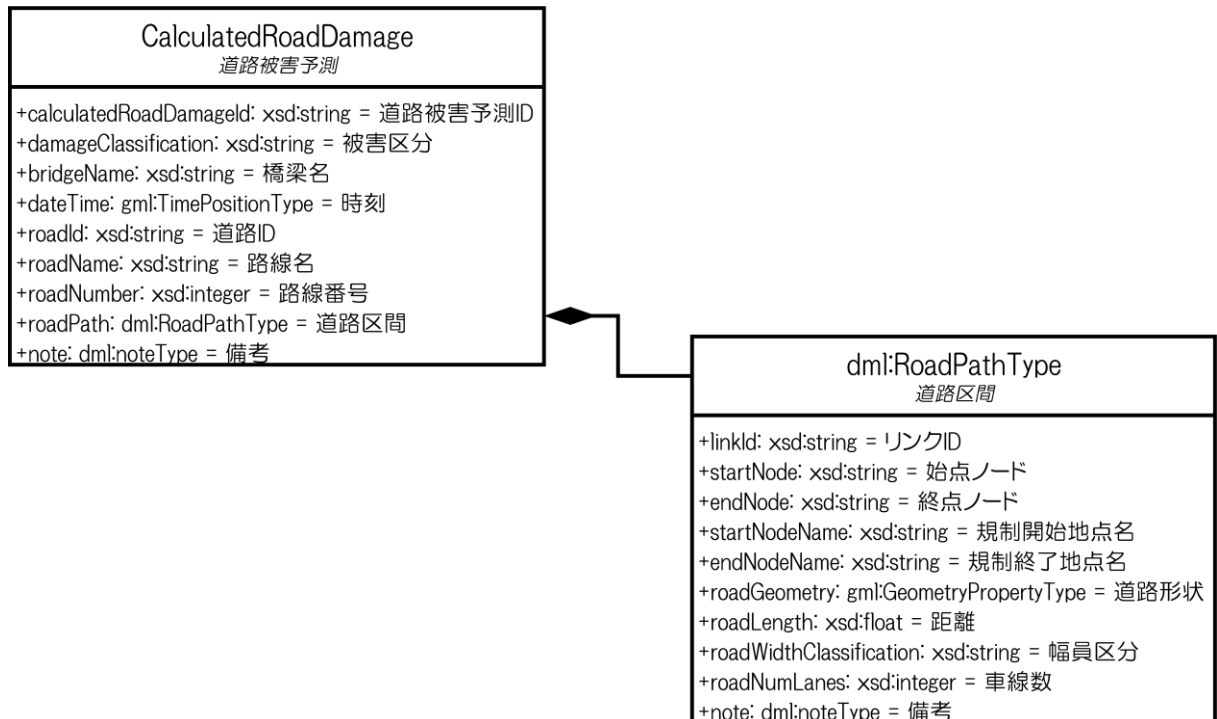


図 1 4 道路被害予測(CalculatedRoadDamage) の構造

```

<CalculatedRoadDamage xmlns="http://www.infosharp.org/schemas/dml">
  <calculatedRoadDamageId>
    CalculatedRoadDamage-2009-12-18-0258
  </calculatedRoadDamageId>
  <damageClassification>Fire</damageClassification>
  <dateTime>2009-12-18T14:00:00+09:00</dateTime>
  <roadId>Road13-14-A-13891</roadId>
  <roadName>409</roadName>
  <roadNumber>409</roadNumber>
  <roadPath>
    <linkId>Link13-14-A-13891</linkId>
    <startNode>533925-13777</startNode>
    <endNode>533925-20396</endNode>
    <roadGeometry>
      <LineString xmlns="http://www.opengis.net/gml">
        <coordinates>
          139.6716427942,35.5583286814
          139.6720807843,35.5576457701
          139.6721677828,35.5574957893
        </coordinates>
      </LineString>
    </roadGeometry>
    <roadLength>104</roadLength>
    <roadWidthClassification>5.5-13.0meter</roadWidthClassification>
  </roadPath>
  <note/>
</CalculatedRoadDamage>
  
```

7) 実証デモまとめ

以上のように、実証デモにおいては、DaRuMa を核にして多種多様なシステムを様々な形で接続・連携し、既存・新規開発を問わず、多くのシステムを柔軟に統合して情報共有を実現できることを示すことができた。

一方、XMLや通信をベースとしたシステム連携は、技術的には可能なものの、対応のコストがかかる場合も予想され、今後はこれらのシステム連携を円滑にする機能やツールの充実と、ガイドラインなどの整備が必要であることが分かった。

8) 国際標準化に関する調査

現在、災害情報の基礎となる位置情報や地理情報については、Open Geospatial Consortium (OGC) や Object Management Group (OMG) においてさまざまな側面から標準化が進められている。本年度はこのなかで、センシングやロボットの技術などへの応用を目指して位置情報の標準化活動を行なっている OMG の技術委員会に参加し、その動向などを調査した。

OMG では数年前より、ユビキタスセンシングやロボットによる自動情報収集における位置・地図情報の取り扱い方の標準化を進めてきている。本会議ではその標準の正式認証が行われ、これから各国でその標準に準拠したシステムの開発が進められることが確認された。本標準には、プロジェクトで開発を進めている減災情報共有データベースのプロトコルである MISP で採用している検索概念を採用しており、本プロジェクトのプラットフォームを比較的容易に標準に準拠させられることがわかった。

また、この標準の形式は汎用性が高いものであるため、単なる位置情報以外に、個人特定情報など情報サービスにも適用が可能であるという議論が進められており、広く災害情報や安全情報への応用が広がると予想される。

この他、同団体では、情報システムのディペンダビリティの標準化に関する議論がはじめられており、各国の事例や取り組みなどが紹介された。ここではあくまでシステムの頑健性が主として議論されているが、その考え方は人間を含めた災害情報システムにも共有するものであり、大いに参考になるものと考えられた。今後とも同標準の動向を調査する必要があると思われる。

(c) 結論ならびに今後の課題

本サブテーマでは、当初の予定通り、前年度までに拡張した DaRuMa による広域連携の評価実証のため、個別研究テーマの他グループで開発するシステムとの連携ツールの開発および改良を進め、12月および3月の2回の実証デモに参画した。また、国際標準化団体の技術委員会での調査を行い、DaRuMa で用いている MISP と国際標準との整合性について働きかけを行った。

一方、実証デモにおいて、既存システムとの連携をより円滑にする必要性が判明し、今後はこれらのシステム連携を円滑にする機能やツールの充実と、ガイドラインなどの整備を進めることが重要であることを確認した。

(d) 引用文献

特に無し

(e) 学会等発表実績

学会等における口頭・ポスター発表
なし

学会誌・雑誌等における論文掲載

掲載論文（論文題目）	発表者氏名	発表場所 （雑誌等名）	発表時期	国際・国内の別
統合シミュレーションと防災	野田五十樹	ロボット情報学ハンドブック(ナノオプティクス・エナジー出版局)	2010.03	国内
Multi-Agent Social Simulation	Itsuki Noda, Peter Stone, Tomohisa Yamashita, Koichi Kurumatani	Handbook of Ambient Intelligence and Smart Environments	2009.12	国際

マスコミ等における報道・掲載

なし

(f) 特許出願, ソフトウェア開発, 仕様・標準等の策定

1) 特許出願

なし

2) ソフトウェア開発

名称	機能
減災情報共有データベース (DaRuMa) ver. 20100331	検索機能強化, 接続機能強化

3) 仕様・標準等の策定

なし

(3) 平成22年度業務計画案

来年度については、本年度までに構築してきた情報共有システムのプラットフォームの普及を進めるため、関係自治体の実情を調査し、広域連携を実装・普及する上で障害となっている問題点や課題を抽出する。これを踏まえて、プラットフォーム利用のためのガイドラインを整備し、加えて各種業務に適用可能な汎用のデータ入力装置など各自自治体の庁内システム連携にむけた機能拡張を行う。